PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

Application Number	09/816,887
Filing Date	March 23, 2001
First Named Inventor	William A. Pugh
Art Unit	2164
Examiner Name	Chojnaki, Mellissa M.
Attorney Docket Number	109870-130111

ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☑ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☑ Other Enclosure(s) (please Identify below):

Return Receipt Postcard

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Schwabe, Williamson & Wyatt, P.C.		
Signature			
Printed name	Robert C. Peck		
Date	June 21 2006	Reg. No.	56,826

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature			
Typed or printed name	Yvette L. Chriscaden	Date	June 21, 2006

Docket No.: 109870-130111 **MAIL STOP: APPEAL BRIEF-PATENTS**

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below.

By: _____ Date: June 21, 2006
Yvette L. Chriscaden

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Before the Board of Patent Appeals and Interferences

App. No.	:	09/816,887	Confirmation No.: 6896
Inventor	:	William A. Pugh	
Filed	:	March 23, 2001	
Title	:	DATABASE AND OPERATING SYSTEM INDEPENDENT COPYING/ARCHIVING OF A WEB BASED APPLICATION	
Art Unit	:	2164	
Examiner	:	Chojnacki, Mellissa M.	
Customer No.	:	25,943	

MAIL STOP: APPEAL BRIEF-PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESUBMISSION OF APPELLANT'S BRIEF IN SUPPORT OF APPELLANT'S APPEAL TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Dear Sir:

This is a re-submission of Appellant's Brief in response to the Non-Compliant Notice mailed on June 2, 2006. The deficiency has been corrected. This appeal furthers the Notice of Appeal filed on October 12, 2005. The appeal arises from a final decision by the Examiner in the final Office Action, dated July 12, 2005. The final decision was in response to arguments filed on April 13, 2005, in response to an earlier office action, mailed January 13, 2005.

Appellants re-submit this *Brief on Appeal* in triplicate. Payment in the amount of $500.00 to cover the fee for filing the *Brief on Appeal* was tendered with the original submission. Appellants respectfully request consideration of this appeal by the Board of Patent Appeals and Interferences for allowance of the present patent application.

Real Party in Interest:

This application is assigned to BEA Systems, Inc., having a principal place of business at 2315 North First Street, San Jose, California 95131. The assignment is recorded at the United States Patent and Trademark Office, reel 014432, frame 0957.

Related Appeals and Interferences:

To the best of Appellants' knowledge, there are no related appeals or interference proceedings currently pending, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

Status of Claims:

Appellants appeal the rejection of claims 1-38. Claims 1-38 were pending and were rejected in the Final Office Action dated July 12, 2005. Claims 1-38 are reproduced, as pending, in Appendix A.

Status of Amendments:

Appellants have made no amendments subsequent to the Examiner's final rejection.

Summary of the Claimed Subject Matter:

Independent claim 1 is directed towards *a method for copying/archiving a web based application* that comprises

> "initializing a file to store said web based application, including creation of a root directory within said file;

creating data directories under said root directory, and initializing a first plurality of storage data objects under said data directories for all non-file system structures of the web based application; and

copying and storing said non-file system structures into said first plurality of storage data objects."

Element 112 of Figure 1 illustrates one example of a copy/archive utility performing the operations recited in claim 1. Element 112 of Figure 1 is described in detail on page 7, lines 3-20, in accordance with some embodiments. Figure 2 illustrates a flowchart of selected operations of the present invention in accordance with claim 1. The selected operations illustrated by Figure 2 are described in greater detail on pages 7, line 23 through page 10, line 18, in accordance with some embodiments. Figure 3 illustrates an exemplary initialized file in accordance with claim 1. The exemplary file is described in greater detail on pages 7, line 23 through page 10, line 18, in accordance with some embodiments.

Independent claim 11 is directed towards *an apparatus* that comprises

"storage medium having stored therein programming instructions, when executed, operate the apparatus to:

initialize a file to store said web based application, including creation of a root directory within said file;

create data directories under said root directory, and initializing a first plurality of storage data objects under said data directories for all non-file system structures of the web based application; and

copy and store said non-file system structures into said first plurality of storage data objects; and

a processor coupled to the storage medium to execute the programming instructions."

Element 112 of Figure 1 illustrates one example of a copy/archive utility capable of serving as the apparatus of claim 11. Element 112 of Figure 1 is described in detail on page 7, lines 3-20, in accordance with some embodiments. Figure 2 illustrates a

flowchart of selected operations of the present invention performed by the apparatus of claim 11. The selected operations illustrated by Figure 2 are described in greater detail on pages 7, line 23 through page 10, line 18, in accordance with some embodiments. Figure 3 illustrates an exemplary initialized file produced by the apparatus of claim 11. The exemplary file is described in greater detail on pages 7, line 23 through page 10, line 18, in accordance with some embodiments. Figure 5 illustrates an exemplary computer system capable of serving as the apparatus of claim 11, in accordance with some embodiments. Figure 5 is described in greater detail on page 14, lines 6-24.

Independent claim 21 is directed towards *a method for copying/restoring a web based application into a domain* that comprises

"retrieving a structural description describing non-file system structures and files of the web based application;

determining in accordance with at least said structural description non-file system structures of the web based application, including constitutions of the non-file system structures, and files of the web based application, including pathnames of the files;

retrieving schemas and data of said non-file system structures in accordance with the result of said determination;

storing said data of said non-file system structures in accordance with schemas of said non-file system structures; and

retrieving and storing said files in accordance with the result of said determination"

Element 112 of Figure 1 illustrates one example of a copy/archive utility performing the copy/restore operations recited in claim 21. Element 112 of Figure 1 is described in detail on page 7, lines 3-20, in accordance with some embodiments. Figure 4 illustrates a flowchart of selected operations of the present invention in accordance with claim 21. The selected operations illustrated by Figure 4 are described in greater detail on pages 10, line 20 through page 14, line 4, in accordance with some embodiments.

Independent claim 23 is directed towards *an apparatus* that comprises

"a storage medium having stored therein a plurality of programming instructions, when executed, operate the apparatus to:

retrieve a structural description describing non-file system structures and files of a web based application,

determine in accordance with at least said structural description non-file system structures of the web based application, including constitutions of the non-file system structures, and files of the web based application, including pathnames of the files,

retrieve schemas and data of said non-file system structures in accordance with the result of said determination,

store said data of said non-file system structures in accordance with schemas of said non-file system structures, and

retrieve and store said files in accordance with the result of said determination; and

at least one processor coupled to the storage medium to execute the programming instructions."

Element 112 of Figure 1 illustrates one example of a copy/archive utility capable of serving as the apparatus of claim 23 by performing the copy/restore operations recited in claim 23. Element 112 of Figure 1 is described in detail on page 7, lines 3-20, in accordance with some embodiments. Figure 4 illustrates a flowchart of selected operations of the apparatus of claim 23. The selected operations illustrated by Figure 4 are described in greater detail on pages 10, line 20 through page 14, line 4, in accordance with some embodiments. Figure 5 illustrates an exemplary computer system capable of serving as the apparatus of claim 23, in accordance with some embodiments. Figure 5 is described in greater detail on page 14, lines 6-24.

Independent claim 25 is directed towards *a method for copying/restoring a web based application into a domain* that comprises

"retrieving a plurality of data table schemas for a plurality of data tables of the web based application, and data of the data tables;

as each data table schema is retrieved,

storing the data table schema in a temporal storage location,

creating a data table in accordance with the data table schema,

determining if data for the data table has already been retrieved,

storing the data into the data table if the data for the data table has already been retrieved; and

as each collection of data for a data table is retrieved,

storing the collection of data in a temporal storage location,

determining if the data table has already been created,

storing the data into the data table if the data table has already been created."

Element 112 of Figure 1 illustrates one example of a copy/archive utility performing the copy/restore operations recited in claim 25. Element 112 of Figure 1 is described in detail on page 7, lines 3-20, in accordance with some embodiments. Figure 4 illustrates a flowchart of selected operations of the present invention in accordance with claim 25. The selected operations illustrated by Figure 4 are described in greater detail on pages 10, line 20 through page 14, line 4, in accordance with some embodiments.

Independent claim 32 is directed towards *an apparatus* that comprises

"a storage medium having stored therein a plurality of programming instructions, when executed, operate the apparatus to:

retrieve a plurality of data table schemas for a plurality of data tables of a web based application, and data of the data tables,

as each data table schema is retrieved,

store the data table schema in a temporal storage location,

create a data table in accordance with the data table schema,

determine if data for the data table has already been retrieved,

store the data into the data table if the data for the data table has already been retrieved, and

as each collection of data for a data table is retrieved,
store the collection of data in a temporal storage location,
determine if the data table has already been created,
store the data into the data table if the data table has already been created; and
at least one processor coupled to the storage medium to execute the programming instructions."

Element 112 of Figure 1 illustrates one example of a copy/archive utility capable of serving as the apparatus of claim 32 by performing the copy/restore operations recited in claim 32. Element 112 of Figure 1 is described in detail on page 7, lines 3-20, in accordance with some embodiments. Figure 4 illustrates a flowchart of selected operations of the apparatus of claim 32. The selected operations illustrated by Figure 4 are described in greater detail on pages 10, line 20 through page 14, line 4, in accordance with some embodiments. Figure 5 illustrates an exemplary computer system capable of serving as the apparatus of claim 32, in accordance with some embodiments. Figure 5 is described in greater detail on page 14, lines 6-24.

Grounds For Rejection To Be Argued On Appeal:

I. Claims 1, 3-5, 9, 11, 13-15, and 19 stand rejected under 35 U.S.C. §103(a) over the teachings of U.S. Patent No. 6,823,338 to *Byrne, et al.* (hereinafter "Byrne") in view of U.S. Patent No. 6,208,993 to *Shadmon* (hereinafter "Shadmon").

II. Claims 6-8, 16-18, 21, 23, 32, and 33-38 stand rejected under 35 U.S.C. §103(a) over the teachings of Byrne in view of Shadmon, as applied to claims 1, 3-5, 9, 11, 13-15, and 19, and further in view of U.S. Patent No. 6,052,693 to *Smith, et al.* (hereinafter "Smith").

III. Claims 25-31 stand rejected under 35 U.S.C. §103(a) over the teachings of Byrne in view of Smith.

IV. Claims 2 and 12 stand rejected under 35 U.S.C. §103(a) over the teachings of Byrne in view of Shadmon, as applied to claims 1, 3-5, 9, 11, 13-15, and 19, further in view of Smith, and further in view of U.S. Patent No. 6,604,106 to *Bodin, et al.* (hereinafter "Bodin").

V. Claims 10, 20, 22, and 24 stand rejected under 35 U.S.C. §103(a) over the teachings of Byrne in view of Shadmon, as applied to claims 1, 3-5, 9, 11, 13-15, and 19, further in view of Smith, and further in view of U.S. Patent No. 6,651,096 to *Gai, et al.* (hereinafter "Gai").

Arguments:

I. Rejection of claims 1, 3-5, 9, 11, 13-15, and 19, under 35 U.S.C. §103(a) was improper because Byrne and Shadmon, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 1, 3-5, 9, 11, 13-15, and 19 is viewed as a whole.

To establish obviousness under 35 U.S.C. § 103, the Examiner must view the invention as a whole. Further, the Examiner is to perform the obviousness analysis in accordance with the standard set forth by the Supreme Court in *Graham v. John Deere Co.*, 383 U.S. 1 (1966). That standard requires that the Examiner (1) determine the scope and content of the prior art; (2) ascertain the differences between the prior art

and the claims in issue; (3) resolve the level of ordinary skill in the art; and (4) evaluate evidence of secondary considerations. *Id.* at 17-18; *see also* MPEP 2141. Secondary considerations include whether the invention met with commercial success, whether the invention answered a long felt need, and whether others attempting the invention have failed. *Graham*, 383 U.S. at 17-18. Further, in applying the *Graham* framework, the Examiner must consider the invention as a whole, without the benefit of hindsight. MPEP 2141.

Claim 1 calls for a method for copying/archiving a web based application including the operations of:

initializing a file to store the web based application, including creation of a root directory within the file;

creating data directories under the root directory, and initializing storage data objects under the data directories for all non-file system structures of the web based application; and

copying and storing the non-file system structures into the storage data objects.

As such, claim 1 describes copying/archiving a web-based application using a file to store the web based application. The file exhibits its own file directory within the file including a root directory and data directories. Non-file system structures of the web based application are stored in the file under the storage data objects.

In contrast, Byrne teaches a "method for securing and processing sparse access control list (ACL) data in a relational database used as a backing store for a hierarchical-based directory service." The directory is made available through an LDAP server, and has a root under which directory entries are organized. Access to the contents of the directory is controlled by a relational database that has tables storing the ACL data. The tables themselves are not part of the directory (see BYRNE Figures 1, 2 , and 5). Thus, they are <u>not stored</u> as objects under the directory, but rather, like a lock

to a safe, are an outside component controlling access to the entries of the directory. Therefore, Byrne fails to teach the operations recited in claim 1 of archiving web based applications in a platform independent file, the file containing a root directory where non-file system objects, such as tables, are stored as storage data objects under a data directory, which in turn is created under the above-mentioned root directory.

In further contrast, Shadmon "teaches a method and a system for uniformly accessing multiple directory services." Shadmon claims an index that is arranged in blocks and is linked to directory items, the items themselves corresponding to data records. The directory taught by Shadmon, however, merely has entries corresponding to data records. It does not, like the present invention as claimed in claim 1, claim a directory storing non-file system structures like data records. Thus, Shadmon and Byrne, taken as a whole, fail to disclose the storing of non-file system structures in a directory.

Additionally, the data dictionary taught by Shadmon simply "maintains meta-data information, which provides information on the data records." In contrast, the present invention as claimed in claim 1 teaches a data directory for storing storage data objects. These storage data objects are claimed to store non-file system structures, such as data records. Thus, unlike the data dictionary of Shadmon which merely maintains information about data records, the data directory of the present invention as claimed in claim 1 actually stores data records.

Thus, Byrne and Shadmon, alone or in combination, fail to teach two features key to the structure of the claimed invention of claim 1: the storing of non-file system structures under a directory, and the creation of a data directory under a root directory. These key features illustrate a novel relationship – between maintaining a directory and storing within that directory the very tables and other non-file system structures to which the directory refers. It is this novel relationship, then, that differentiates the present

invention as claimed in claim 1 from Byrne and Shadmon. Consequently, claim 1 is patentable over Byrne and Shadmon.

Claim 11 contains in substance the same recitations earlier discussed for claim 1. Therefore, for at least the same reasons, claim 11 is patentable over Byrne and Shadmon.

Claims 3-5, 9, 11, 13-15, and 19 depend from either claim 1 or 11, incorporating their limitations correspondingly. Accordingly, for at least the same reasons, claims 3-5, 9, 13-15, and 19 are patentable over Byrne and Shadmon.

II. Rejection of claims 6-8, 16-18, 21, 23, 32, and 33-38, under 35 U.S.C. §103(a) was improper because Byrne, Shadmon, and Smith, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 6-8, 16-18, 21, 23, 32, and 33-38 is viewed as a whole.

Smith does not remedy the above-discussed deficiencies of Byrne and Shadmon. Therefore, claims 1, 11, 21, 23, and 32 remain patentable over Byrne and Shadmon even when combined with Smith.

Claims 6-8, 16-18, and 33-38 depend on claims 1, 11, and 32, incorporating their limitations respectively. Therefore, for at least the same reasons, Claims 6-8, 16-18, 21, 23, 32, and 33-38 are patentable over Byrne, Shadmon, and Smith, alone or in combination.

III. Rejection of claims 25-31 under 35 U.S.C. §103(a) was improper because Byrne and Smith, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 25-31 is viewed as a whole.

Smith does not remedy the above-discussed deficiency of Byrne. Therefore, claim 25 remains patentable over Byrne and Smith, alone or in combination.

Claims 26-31 depend on claim 25, incorporating its limitations respectively. Therefore, for at least the same reasons, claims 26-31 are patentable over Byrne and Smith, alone or in combination.

IV. Rejection of claims 2 and 12, under 35 U.S.C. §103(a) was improper because Byrne, Shadmon, Smith, and Bodin, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 2 and 12 is viewed as a whole.

Bodin and Smith, alone or in combination, do not remedy the above-discussed deficiencies of Byrne and Shadmon. Therefore, claims 1 and 11 remain patentable over Byrne, Shadmon, Smith, and Bodin, alone or in combination.

Claims 2 and 12 depend on either Claims 1 or 11, incorporating their limitations respectively. Therefore, for at least the same reasons, Claims 2 and 12 are patentable over Byrne, Shadmon, Smith, and Bodin, alone or in combination.

V. Rejection of claims 10, 20, 22, and 24, under 35 U.S.C. §103(a) was improper because Byrne, Shadmon, Smith, and Gai, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 10, 20, 22, and 24 is viewed as a whole.

Smith and Gai, alone or in combination, do not remedy the above-discussed deficiencies of Byrne and Shadmon. Therefore, claims 1, 11, 21, and 23 remain patentable over Byrne, Shadmon, Smith, and Gai, alone or in combination.

Claims 10, 20, 22, and 24 depend on claims 1, 11, 21, and 23, incorporating their limitations respectively. Therefore, for at least the same reasons, Claims 10, 20, 22, and 24 are patentable over Byrne, Shadmon, Smith, and Gai, alone or in combination.

Conclusion

Appellant respectfully submits that all the appealed claims in this application are patentable and requests that the Board of Patent Appeals and Interferences overrule the Examiner and direct allowance of the rejected claims.

The fees associated with the appeal brief were submitted with the original appeal brief. We do not believe any additional fees, in particular extension of time fees, are needed. However, should that be necessary, please charge our deposit account 500393. In addition, please charge any shortages and credit any overages to Deposit Account No. 500393.

Respectfully submitted,

Date: June 21, 2006

Robert C. Peck, Reg. No. 56,826
Agent for Appellant Applicants

Schwabe Williamson & Wyatt, P.C.
1420 Fifth, Suite 3010
Seattle, WA 98101
Tel: (206) 622-1711
Fax: (206) 292-0460

Appendix A – Appealed Claims

1. (Original) A method for copying/archiving a web based application, said method comprising:

initializing a file to store said web based application, including creation of a root directory within said file;

creating data directories under said root directory, and initializing a first plurality of storage data objects under said data directories for all non-file system structures of the web based application; and

copying and storing said non-file system structures into said first plurality of storage data objects.

2. (Previously Presented) The method of claim 1, wherein said initializing of a file to store said web based application comprises initializing a compressible file.

3. (Original) The method of claim 1, wherein said creating of data directories under said root directory and initializing a first plurality of storage data objects under said data directories comprises creating an application level data directory under said root directory.

4. (Previously Presented) The method of claim 3, wherein said creating of data directories under said root directory and initializing a first plurality of storage data objects under said data directories further comprises:

initializing a first of said first plurality of storage data objects under said application level data directory to store a structural description describing non-file system structures and files of a file system of the web based application; and

copying and storing said structure description in said first of said first plurality of storage data objects.

5. (Previously Presented) The method of claim 4, wherein said copying and storing of non-file system structures into said first plurality of storage data objects comprises:

initializing a second of said first plurality of storage data objects under said application level data directory to store a user description describing users of the web based application; and

copying and storing said user description in said second of said first plurality of storage data objects.

6. (Original) The method of claim 3, wherein said creating of data directories under said root directory and initializing a first plurality of storage data objects under said data directories further comprises creating a plurality of data table directories under said application level data directory.

7. (Previously Presented) The method of claim 6, wherein said creating of data directories under said root directory and initializing a first plurality of storage data objects under said data directories further comprises:

initializing a first subset of said first plurality of storage data objects under said data table directory to store data table schemas of the web based application; and

initializing a second subset of said first plurality of storage data objects under said data table directory to data tables of the web based application.

8. (Original) The method of claim 7, wherein said copying and storing of non-file system structures into said first plurality of storage data objects comprises copying and storing data table schemas and data tables of the web based application into corresponding pairs of said first and second subset of said first plurality of storage data objects.

9. (Original) The method of claim 1, wherein the method further comprises copying and storing files of the web based application that are part of a file system into

said file for storing said web based application as second plurality of storage data objects under said root directory.

10. (Original) The method of claim 9, wherein said copying and storing of files of the web based application that are part of a file system into said file for storing said web based application as second plurality of storage data objects under said root directory comprises pre-processing access control lists into a self-describing format before storing the access control lists into selected ones of said second plurality of storage data objects.

11. (Previously Presented) An apparatus comprising:

storage medium having stored therein programming instructions, when executed, operate the apparatus to:

initialize a file to store said web based application, including creation of a root directory within said file;

create data directories under said root directory, and initializing a first plurality of storage data objects under said data directories for all non-file system structures of the web based application; and

copy and store said non-file system structures into said first plurality of storage data objects; and

a processor coupled to the storage medium to execute the programming instructions.

12. (Previously Presented) The apparatus of claim 11, wherein said programming instructions, when executed, operate the apparatus to initialize a compressible file to store said web based application.

13. (Original) The apparatus of claim 11, wherein said programming instructions, when executed, operate the apparatus to create an application level data directory under said root directory to create data directories under said root directory and initialize a first plurality of storage data objects under said data directories.

14. (Previously Presented) The apparatus of claim 13, wherein said programming instructions, when executed, operate the apparatus to:

initialize a first of said first plurality of storage data objects under said application level data directory to store a structural description describing non-file system structures and files of a file system of the web based application, and

copy and store said structure description in said first of said first plurality of storage data objects.

15. (Original) The apparatus of claim 14, wherein said programming instructions, when executed, operate the apparatus to initialize a second of said first plurality of storage data objects under said application level data directory to store a user description describing users of the web based application, and to copy and store said user description in said second of said first plurality of storage data objects.

16. (Original) The apparatus of claim 13, wherein said programming instructions, when executed, operate the apparatus to create a plurality of data table directories under said application level data directory to create data directories under said root directory and initialize a first plurality of storage data objects under said data directories.

17. (Previously Presented) The apparatus of claim 16, wherein said programming instructions, when executed, operate the apparatus to:

initialize a first subset of said first plurality of storage data objects under said data table directory to store data table schemas of the web based application, and

initialize a second subset of said first plurality of storage data objects under said data table directory to data tables of the web based application.

18. (Original) The apparatus of claim 17, wherein said programming instructions, when executed, operate the apparatus to copy and store data table schemas and data

tables of the web based application into corresponding pairs of said first and second subset of said first plurality of storage data objects to copy and store non-file system structures into said first plurality of storage data objects.

19. (Original) The apparatus of claim 11, wherein said programming instructions, when executed, operate the apparatus to copy and store files of the web based application that are part of a file system into said file for storing said web based application as second plurality of storage data objects under said root directory.

20. (Original) The apparatus of claim 19, wherein said programming instructions, when executed, operate the apparatus to pre-process access control lists into a self-describing format before storing the access control lists into selected ones of said second plurality of storage data objects.

21. (Previously Presented) A method for copying/restoring a web based application into a domain, said method comprising:

retrieving a structural description describing non-file system structures and files of the web based application;

determining in accordance with at least said structural description non-file system structures of the web based application, including constitutions of the non-file system structures, and files of the web based application, including pathnames of the files;

retrieving schemas and data of said non-file system structures in accordance with the result of said determination;

storing said data of said non-file system structures in accordance with schemas of said non-file system structures; and

retrieving and storing said files in accordance with the result of said determination.

22. (Original) The method of claim 21, wherein said retrieving and storing of files of the web based application comprises transforming one or more access control lists into a binary format before storing the one or more access control lists.

23. (Previously Presented) An apparatus comprising:

a storage medium having stored therein a plurality of programming instructions, when executed, operate the apparatus to:

retrieve a structural description describing non-file system structures and files of a web based application,

determine in accordance with at least said structural description non-file system structures of the web based application, including constitutions of the non-file system structures, and files of the web based application, including pathnames of the files,

retrieve schemas and data of said non-file system structures in accordance with the result of said determination,

store said data of said non-file system structures in accordance with schemas of said non-file system structures, and

retrieve and store said files in accordance with the result of said determination; and

at least one processor coupled to the storage medium to execute the programming instructions.

24. (Original) The apparatus of claim 23, wherein the programming instructions, when executed, further operate the apparatus to transform an access control list into a binary format before storing the access control list.

25. (Original) A method for copying/restoring a web based application into a domain, said method comprising:

retrieving a plurality of data table schemas for a plurality of data tables of the web based application, and data of the data tables;

as each data table schema is retrieved,

storing the data table schema in a temporal storage location,

creating a data table in accordance with the data table schema,

determining if data for the data table has already been retrieved,

storing the data into the data table if the data for the data table has already been retrieved; and

as each collection of data for a data table is retrieved,

storing the collection of data in a temporal storage location,

determining if the data table has already been created,

storing the data into the data table if the data table has already been created.

26. (Original) The method of claim 25, wherein the method further comprises upon storing the data of a data table into the data table, deleting the data table schema and the data of the data table stored in the respective temporal storage locations.

27. (Original) The method of claim 25, wherein the method further comprises deleting log-in user names of users when storing data into a data table if the data table is an address book.

28. (Original) The method of claim 27, wherein the method further comprises determining if users having entries in an address book are authorized to log in the domain, and adding into corresponding entries of the address book log-in user names of users authorized to log in the domain.

29. (Original) The method of claim 25, wherein the method further comprises conditionally deleting or retaining log-in user names of users depending on whether the users are authorized to log in the domain when storing data into a data table if the data table is an address book.

30. (Previously Presented) The method of claim 25, wherein the method further comprises:

retrieving a list of users of the web based applications;

determining if the users are registered with the domain; and

registering the users with the domain if the users are determined to be not having registered with the domain.

31. (Previously Presented) The method of claim 30, wherein the method further comprises:

determining if the users already have corresponding entries in an address book of the web based application;

creating the corresponding entries in the address book if the corresponding entries are determined not to have been previously created; and

upon either determining the existence or creation of the corresponding entries, updating the corresponding entries with log-in user names of the users.

32. (Previously Presented) An apparatus comprising:

a storage medium having stored therein a plurality of programming instructions, when executed, operate the apparatus to:

retrieve a plurality of data table schemas for a plurality of data tables of a web based application, and data of the data tables,

as each data table schema is retrieved,

store the data table schema in a temporal storage location,

create a data table in accordance with the data table schema,

determine if data for the data table has already been retrieved,

store the data into the data table if the data for the data table has already been retrieved, and

as each collection of data for a data table is retrieved,

store the collection of data in a temporal storage location,

determine if the data table has already been created,

store the data into the data table if the data table has already been created; and

at least one processor coupled to the storage medium to execute the programming instructions.

33. (Original) The apparatus of claim 32, wherein the programming instructions, when executed, further operate the apparatus to delete the data table schema and the data of the data table stored in the respective temporal storage locations, upon storing the data of a data table into the data table.

34. (Original) The apparatus of claim 32, wherein the programming instructions, when executed, further operate the apparatus to delete log-in user names of users when storing data into a data table if the data table is an address book.

35. (Original) The apparatus of claim 34, wherein the programming instructions, when executed, further operate the apparatus to determine if users having entries in an address book are authorized to log in the domain, and add into corresponding entries of the address book log-in user names of users authorized to log in the domain.

36. (Original) The apparatus of claim 32, wherein the programming instructions, when executed, further operate the apparatus to conditionally delete or retain log-in user names of users depending on whether the users are authorized to log in the domain when storing data into a data table if the data table is an address book.

37. (Previously Presented) The apparatus of claim 32, wherein the programming instructions, when executed, further operate the apparatus to:

retrieve a list of users of the web based applications;

determine if the users are registered with the domain; and

register the users with the domain if the users are determined to be not having registered with the domain.

38. (Previously Presented) The apparatus of claim 37, wherein the programming instructions, when executed, further operate the apparatus to:

determine if the users already have corresponding entries in an address book of the web based application;

create the corresponding entries in the address book if the corresponding entries are determined not to have been previously created; and

upon either determining the existence or creation of the corresponding entries, update the corresponding entries with log-in user names of the users.

Appendix B – Copies of Evidence Submitted

No evidence has been submitted under 37 C.F.R. 1.130, 1.131, or 1.132. No evidence entered by Examiner has been relied upon by Appellants in the appeal.